

# H IMM을 이용한 3차원 지도작성

정현룡\*(전남대 대학원 기계공학과), 김영배\*\*

## 3D Map-Building using Histogramic In-Motion Mapping in the Eyebot

Jung, HyunRyong(Mech. Eng. Dept, CNU), YoungBac Kim

### ABSTRACT

This paper introduces histogramic in-motion mapping for real-time map building with the Eyebot in motion. A histogram grid used in H IMM is updated through three PSD sensors. H IMM makes it possible to make fast map-building and avoid obstacles in real-time. Fast map-building allows the robot to immediately use the mapped information in real-time obstacle-avoidance algorithms.

H IMM has been tested on the Eyebot. The Eyebot sends PSD data to computer and computer builds a 3D-Map based on PSD data.

**Key Words** : H IMM(Histogramic In-Motion Mapping), Map-building, CV, GRO, OpenGL, Cell grid

### 1. 서론

현재 우리 주위에는 소니(Sony)의 아이보, 혼다(Honda)의 아시모등이 다가오고 있다. 이러한 로봇이 작동하기 위해서는 많은 기능들을 필요로 한다. 그 중 가장 중요한 기능은 바로, Map-building과 Path Finding이라고 할 수 있다. 이러한 기능이 없다면, 로봇의 이동이 불가능하다. 이 논문에서는 로봇에게 필수 불가결한 기능 중에 하나인 Map-Building에 대한 것이다.

최근 들어 Map-Building에 스테레오 비전을<sup>[1]</sup> 이용하는 방법이 있으나 시스템의 고 사양을 요구하고, 속도가 느리다는 단점이 있다. 본 논문은 3개의 PSD 센서로부터 들어오는 정보를 받아들여, 3차원으로 Map-Building을 하는 방법에 대해 소개한다.

Map Building을 하기 위해서 사용된 로봇은 UWA(University of Western Australia)의 CHIPS Lab에서 개발된 Eyebot을 사용하였다. Eyebot은 two-wheel이며, 3개의 PSD sensor, CMOS camera, digital compass, LCD 그리고 RF module로 이루어져 있다.

Eyebot은 동작하는 동안 현재의 위치정보와 3개의 PSD sensor로부터 거리 정보를 computer로 전송하게 된다. 그리고 전송된 데이터를 바탕으로 computer에서는 OpenGL을 이용하여 3D-Map을 작성한다. 이번 논문에서는 obstacle avoidance보다는 Map-Building에 초점을 맞추도록 하겠다.

### 2. 이론

## 2.1 Eyebot

이 논문에서 사용된 Eyebot은 CHIPS Lab에서 개발된 로봇이다. 로봇의 크기는 가로 15.5cm, 세로 19cm이다. 프로그램 상에서 로봇을 나타내는 셀의 크기는 20cm×20cm으로 표현한다. 로봇은 두 개의 엔코더가 내장된 DC모터가 장착되어 있다. Localization은 dead reckon이 사용되었다. 로봇의 좌표축은 우리가 사용하는 좌표축과 다르기 때문에, 로봇으로부터 받은 값을 변화하여 주어야만 한다. 그러므로 로봇의 x값은 프로그램 상에서는 y값으로 그리고 로봇의 y값은 -x로 바뀌어 사용해야만 한다.

## 2.2 Map Building with PSD Sensors

HMM은 장애물을 표시하는데 있어서 이차원 Cartesian histogram grid를 사용한다. 그리고 histogram grid의 각 cell들은 certainty value(CV)값을 갖는다.

Histogram grid는 갱신되는 방법에 있어서 CMU의 certainty grid method<sup>(2,3,4)</sup>와는 다르다. CMU의 방법은 초음파 센서 영역내의 cell들 모두 갱신시키는 방법을 택하고 있다. 하지만 이러한 방법은 막대한 양의 계산 양을 요구하게 되고 실시간으로 map-building하는 것을 어렵게 만들게 한다. 반면에 HMM은 센서에 의해서 검출된 거리 d에 위치하는 cell의 값만을 증가시키게 된다. 계산 양이 줄어들게 됨으로 로봇이 움직이는 동안 실시간으로 계속해서 map-building이 가능하게 된다. 그리고 센서에 의해서 선택된 cell의 CV값이 증가함과 동시에 주변의 cell들의 CV값도 동시에 증가하게 된다. 하지만, 주의해야 하는 점은 HMM은 로봇의 정지해 있는 경우에는 로봇이 움직이는 경우보다 정확도가 떨어진다는 점이다.

센서에 의해서 선택된 cell의 증가치( $I^+$ )는 3이며, CV의 최대 값( $CV_{max}$ )은 15이다. 그리고 선택된 cell과 센서사이에 놓인 모든 cell들은  $-1(I^-)$  만큼 감소하게 된다(Fig. 1). CV의 최소값( $CV_{min}$ )은 0이다.  $I^+$ 와  $I^-$ 의 값은 각각 독립적으로 결정되게 된다.  $I^+$ 의 값은 실험으로

결정하게 되며,  $I^-$ 의 값은  $I^+$ 보다 작은 값으로 결정하여야 한다. 한번의 센서 reading으로 하나의 cell을 증가하게 되지만, 다수의 cell들은 감소하게 되기 때문이다.

Eyebot에는 좌, 우 그리고 전방에 각각 1개씩의 PSD sensor가 부착되어 있다. PSD sensor가 측정할 수 있는 거리는 7cm에서 80cm까지 이다.

HMM은 원래 초음파 센서를 이용해 Map Building을 하는 알고리즘이다. 하지만, Eyebot에서 사용하는 센서는 PSD이기 때문에, 원래의 이론을 수정하여서 사용하였다.

HMM은 초음파를 사용한다는 가정 하에 이루어지기 때문에, 초음파가 진행하는 부채꼴 모양의 영역에 위치하는 장애물에 의해서 영향을 받게 된다. 즉 초음파가 진행한 영역의 오른쪽에 있거나 왼쪽에 있거나 측정된 거리는 같게 된다. 그래서 정확히 장애물이 어디에 있는지는 알 수 없다. 그러나 PSD의 경우는 초음파처럼 방향성에 있어서 부정확하지 않다.

HMM에서는 로봇이 움직이는 동안에 정확한 map-building을 하기 위해서 Growth Rate Operator(GRO)을 사용한다. GRO는 센서에 의해서 검출된 cell 주변에 CV값이 0보다 큰 cell들이 있을 경우는 현재 선택된 cell에 영향을 주도록 한다. 즉 선택된 cell의 주변 cell들에 CV값이 높으면, 현재 선택된 cell의 CV값도 높아지게 되는 것이다.

GRO를 사용함으로써 로봇이 움직이는 동안 발생할 수 있는 읽기 오류로 인한 잘못된 map-building을 피할 수 있게 된다. 또한 로봇이 빠르게 움직이는 동안 적은 수의 sampling 만으로도 map-building이 가능하게 된다. CV값의 계산 알고리즘을 표현하여 보면 아래 식과 같다.

$$CV'_{i,j} = CV_{i,j} + I^+ + \sum_{n=0}^{P_{i,j}-1} (w_{n,q} CV_{i,j+n,q}) \quad (1)$$

여기서,

$CV_{i,j}$  : previous certainty value of cell  
(i, j)

$CV'_{i,j}$  : updated certainty value of cell

(i, j)

$I^+$  : constant increment ( $I^+ = 3$ )

w : weighting factor

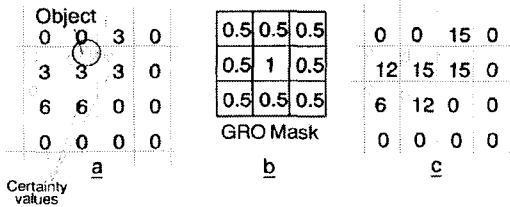


Fig. 1 (a). 로봇이 움직이는 동안 하나의 장애물에 대한 CV값  
 (b). 3×3 mask for growth rate operator(GRO)  
 (c). GRO를 적용한 상태

Table 1은 Fig.1를 기초로한 CV값 계산 예이다.

**2.3 3D Map-building**

Computer에서는 로봇으로부터 오는 거리 데이터와 위치 데이터를 바탕으로 3D Map-Building을 하게 된다. 차후에 적용할 A\* Algorithm을 위해서 로봇의 위치는 cell grid 방식을 취하였다.

Read			CV
1	$a=a+I^+$	$=0+3$	$=(3)$
2	$a=a+I^+$	$=3+3$	$=6$
3	$b=b+I^++\frac{1}{2}(a)$	$=0+3+3$	$=(6)$
4	$b=b+I^++\frac{1}{2}(a)$	$=6+3+3$	$=12$
5	$c=c+I^++\frac{1}{2}(a+b)$	$=0+3+9$	$=12$
6	$d=d+I^++\frac{1}{2}(a+b+c)$	$=0+3+15$	$=15$
7	$e=e+I^++\frac{1}{2}(b+d)$	$=0+3+13.5$	$=15$
8	$f=f+I^++\frac{1}{2}(e+f)$	$=0+3+15$	$=15$

Table 1 Fig4의 CV 계산 예(CVmax = 15, ()의 값은 읽시 값)

로봇의 위치를 나타내는 cell grid의 크기는 로봇의 회전하는 동안 주변 장애물에 접촉하지 않도록 하기 위해서 로봇의 크기 보다 더 큰 정사각형 (20cm × 20cm)으로 정하였다. 로봇으로부터 들어오는 위치 좌표 값은 프로그램 상에서 3D로 표

시되게 된다. OpenGL에서 로봇의 렌더링은 먼저 3D-Max에서 모델링을 한 후 MD2파일로 변환하여 읽어 들이는 방법을 사용하였다<sup>(5)</sup>. 로봇으로부터 들어오는 위치 데이터는 로봇의 VCP(Vehicle Center Point)로 간주하여 화면에 표시하도록 하였다.

**3. 실험**

실험은 computer에서 로봇의 현재 위치 값과 목적지 위치 값을 입력한 후, 그 값을 로봇으로 전송하게 된다. 그리고 로봇은 움직이는 동안 계속해서 거리 데이터와 현재 위치 데이터 값을 전송하게 된다. Table 2는 로봇과 computer간의 데이터를 주고받는 순서이다.

그리고 로봇이 target point로 도착한 경우에는 computer로부터 종료를 알리는 패킷을 Eyebot으로 보내게 된다.

실제 테스트에서 장애물을 표시하는 cell grid의 크기는 2cm×2cm와 4cm×4cm에서 테스트하였다. 그리고 장애물을 나타내는 박스는 CV값에

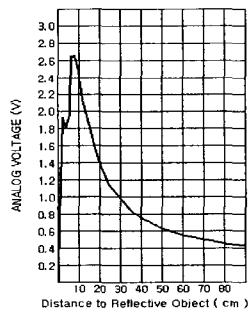


Fig. 2 Property of the PSD

따라서 색깔을 달리하였다(Table 3). 이렇게 CV값에 따라 색깔을 다르게 함으로 보다 쉽게 장애물이 존재할 확률이 높은 위치를 확인할 수 있게 된다.

다음은 4cm×4cm로 cell grid를 정의하고 테스트한 결과이다.

Fig.3의 왼쪽은 3차원으로 분 화면이고, 오른쪽은 2차원으로 분

결과이다. 투명하게 보이는 로봇의 형상은 로봇이 지나온 자취를 저장한 후 보여주도록 하였다. 전체적으로 Map-building이 잘 이루어 졌음을 알 수 있다. 하지만 왼쪽의 장애물이 제대로 그려지지 않았음을 알 수 있다

이러한 이유는 로봇에 장착된 PSD의 거리에 따른

전압의 관계가 Fig.2와 같기 때문에 7cm이하의 거리가 감지된 경우는 잘못된 거리 값이 들어오게 된다. 또한, PSD센서가 전방과 좌, 우에만 위치해 있는 관계로 로봇이 동작하는 동안 PSD센서가 감출하지 못하는 부분이 있음을 알 수 있다. 즉, PSD 센서의 직외선이 도달하지 않는 부분에 대해서는 Map-Building이 이루어지지 않고 있음을 알 수 있다.

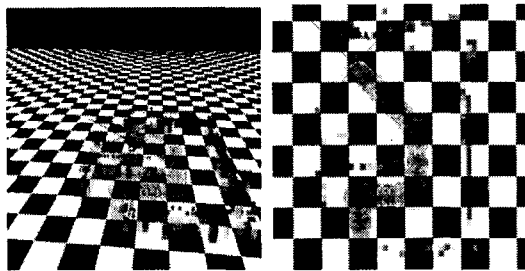


Fig. 3 Experiment(1)

이러한 부분들을 제외하고는 실제 환경과 일치하는 Map-building이 이루어졌음을 알 수 있다.

다음은 장애물을 나타내는 cell grid의 크기를 2cm×2cm로 설정하고 테스트한 결과이다.

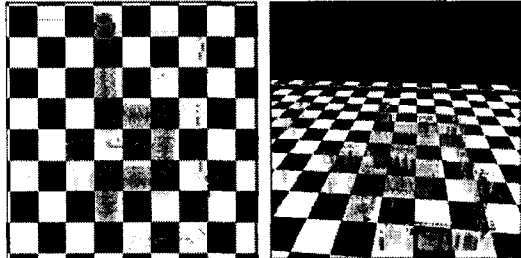


Fig. 4 Experiment(2)

Fig.4에서 보듯이 cell grid의 크기가 작아짐으로써 map을 보다 정밀하게 그릴 수 있었으나, 장애물을 표현하기 위한 cell grid의 수가 증가하게 되어 시스템이 느려지는 결과를 낳았다. 그리고 이번 실험에서도 역시 PSD의 특성으로 인하여 맵이 잘 못 그려지는 부분들이 발견되었다. 이러한 부분들은 로봇과의 거리가 7cm이상일 때는 제대로 그려졌으나, 로봇이 장애물에 접근하면서 왜곡되는 모습을 보였다. 하지만, 로봇이 장애

물에 7cm 이하로 근접하지 않았을 경우에는 좋은 결과를 보여 주었다.

#### 4. 결론

실험에서도 보여 지듯이 HMM은 실시간으로 Map-Building을 할 수 있는 좋은 방법이다. 초기에 나온 HMM은 초음파에 적용되었으나, 이번 논문에서는 PSD센서에 적용하여 좋은 결과를 보여 주었으며, 3개의 PSD센서만을 사용하여도 Map-Building이 가능하다는 것을 알 수 있었다. 장애물을 표시하는 cell grid의 크기는 4cm×4cm보다는 2cm×2cm의 크기가 더 정확히 Map-Building을 할 수 있었다. 또한 2차원 Map정보를 이용해서 3차원화 하면서, Map에 대한 가독성을 높였다. 하지만, 로봇이 장애물에 너무 가까이 접근하는 경우나, 장애물이 멀리 떨어져 있는 경우 PSD 센서에 노이즈가 많이 입력되는 경우는 Map-Building이 잘 이루어지지 않았다. 또한 로봇의 전방에 센서가 하나만 설치되어 있는 관계로 로봇이 회전 이동을 하지 않는 경우는 전방에 대한 Map-Building이 잘 이루어지지 않았다. 하지만, 이러한 문제점은 PSD센서를 전방에 추가 설치하거나, 모터를 사용하여 전방을 스캔하는 방법으로 해결 할 수 있을 것이다.

#### 참고문헌

1. <http://www.dis.uniroma1.it/~iocchi/stereo/stereo.html>
2. Elfes, A, "Using Occupance Grids for Mobile Robot Perception and Navigation." Computer Magazine, June, pp. 46-57 1989
3. Moravec, H. P. and Elfes, A., "High Resolution Maps from Wide Angle Sonar." Proceedings of the IEEE Conference on Robotics and Automation, pp.116-121, 1985
4. Moravec, H. P., "Sensor Fusion in Certainty Grids for Mobile Robots." AI Magazine, Summer, pp.61-74, 1988.
5. Andre LaMothe, "OpenGL. Game Programming", pp.626-675, 2001